

Пусть $X_i = \beta_1 + i\beta_2 + \varepsilon_0 + \dots + \varepsilon_i$, $i = 0, 1, \dots, n$ — расстояния, которое проехал трамвай за i секунд по показанию датчика. Здесь β_1 — начальное расстояние, β_2 — скорость трамвая, ε_0 — ошибка начального показания датчика. Трамвай едет с постоянной скоростью, и через каждую секунду датчик фиксирует расстояние, которое проехал трамвай. Отсчет времени идет от предыдущего замера, причем отсчет происходит с ошибкой. Для $i = 1, \dots, n$ величина ε_i есть ошибка приращения расстояния, то есть $\varepsilon_i = \varepsilon_i^t \beta_2$, где ε_i^t — ошибка отсчета времени. Все ошибки ε_i независимы и распределены по закону $N(0, \sigma^2)$. Сведите задачу к линейной модели и найдите оценки наименьших квадратов для начального расстояния β_1 и скорости β_2 , а также несмещенную оценку для σ^2 , из которой выразим оценку дисперсии отсчета времени.

```
In [30]: import scipy.stats as stats
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import pylab
import seaborn as sns
%matplotlib inline
```

```
In [36]: data = pd.read_csv('Regression.csv')
data[:5]
```

```
Out[36]:
```

	104.9407
0	118.0170
1	133.1054
2	143.4023
3	155.9753
4	171.7617

```
In [37]: data=data.values
```

```
In [40]: data_list = [104.9407]
         for i in range(999):
             data_list.append(float(data[i]))
         len(data_list)
```

Out[40]: 1000

Сводим задачу к линейной модели, как в теоретической задаче 9.2. Введем случайную величину Y : $Y_0 = X_0, Y_1 = X_1 - X_0, Y_2 = X_2 - X_1$ и так далее, получим, что тогда вектор ошибок будет распределен нормально. Получим $(X_0, \frac{X_N - X_0}{n}) = (\beta_1^*, \beta_2^*)$,

$\sigma^{2*} = \frac{1}{n-1}(\sum_{i=1}^n (X_i - X_{i-1})^2 - \frac{(X_n - X_0)^2}{n})$ - несмещенная.

```
In [48]: def do_task():
         n = len(data_list)
         beta_1 = data_list[0]
         beta_2 = (data_list[n - 1] - data_list[0]) / n
         diffs = np.array([data_list[0] if i == 0 else data_list[i] - data_list[i - 1] for i in range(n)])
         sig_2 = (1. / (n - 1)) * (np.sum(diffs ** 2) - (data_list[n-1] - data_list[0])**2 / n)
         dispers_time = sig_2 / beta_2**2
         return [beta_1, beta_2, sig_2, dispers_time]
```

```
In [49]: beta_1, beta_2, sig_2, dispers_time = do_task()
```

```
In [50]: print("Beta_1:", beta_1, "Beta_2:", beta_2, "Sigma^2: ", sig_2, "Time dispersion: ", dispers_time)
```

Beta_1: 104.9407 Beta_2: 13.9189515 Sigma^2: 13.429290934 Time dispersion: 0.0693170448485

Вывод. Мы свели задачу к линейной модели. Получили оценки: β_1 -начальное положение трамвая 104.9407, β_2 - скорость трамвая, около 14, ошибка приращения расстояния ~ 13.4, оценка дисперсии отсчета времени ~ 0.07.

In []: